Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L8	39	(xlock or x?lock or (exclusive adj lock)) with delet$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/03/30 13:12
L9	19	L8 and (deadlock$3 or dead?lock$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/03/30 12:59
L10	11	(xlock or x?lock or (exclusive near1 lock$3)) and (pseudodelet$3 or pseudo?delet$3 or (soft adj delet$3) or (delete near1 (flag or bit))) and (deadlock$3 or dead?lock$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/03/30 13:29
L11	1	(xlock or x?lock or (exclusive near1 lock$3)) and ((xlock or x?lock or lock) near1 delete near1 (attribute or bit or flag))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/03/30 13:18
L12	7	(xlock or x?lock or (exclusive near1 lock$3)) and ((xlock or x?lock or lock) near1 (attribute or bit or flag) near1 delete)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/03/30 13:25
L13	7	(xlock or x?lock or (exclusive near1 lock$3)) and ((xlock or x?lock or lock) near2 (attribute or bit or flag) near2 delete)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/03/30 13:26
L14	9	(xlock or x?lock or (exclusive near1 lock$3)) and ((xlock or x?lock or lock) near3 (attribute or bit or flag) near3 delete)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/03/30 13:28
L15	0	(xlock or x?lock or (exclusive near1 lock$3)) and ((xlock or x?lock or lock) near3 (attribute or bit or flag) near3 delete)	DERWENT	OR	ON	2005/03/30 13:29
L16	0	(xlock or x?lock or (exclusive near1 lock$3)) and (pseudodelet$3 or pseudo?delet$3 or (soft adj delet$3) or (delete near1 (flag or bit))) and (deadlock$3 or dead?lock$3)	DERWENT	OR	ON	2005/03/30 13:29
L18	839	707/8.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/03/30 13:30
L19	39	L18 and (pseudodelet$3 or pseudo?delet$3 or (soft near1 delet$3) or (delet$3 near1 (attribute or flag or bit)))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/03/30 13:32

L20	33	L19 and @ad<="20010628"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/03/30 13:32
S1	3	("20020143732" "20020091694" "20020029209" "6604097").did.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/01 14:41
S2	7	database and index and (key or (unique adj (id or identifier or identification))) and ((exclusive$2 near3 lock$3) or x?lock) and ((pseudo?delet$3 or "pseudo delete") or (lock with delet$3 with (attribute or bit or flag or bool or boolean)))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/01 14:51
S3	7	(database and index and (key or (unique adj (id or identifier or identification))) and ((exclusive$2 near3 lock$3) or x?lock) and ((pseudo?delet$3 or "pseudo delete") or (lock with delet$3 with (attribute or bit or flag or bool or boolean)))) and @ad<="20010628"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/01 15:37
S4	6	database and index and ((exclusive$2 near3 lock$3) or x?lock) and ((shared near3 lock$3) or s?lock) and ((pseudo?delet$3 or pseudodelet$3 or (pseudo adj delet$3) or (soft adj delet$3) or (delet$3 near2 (flag or flagged or flagging or bit or bool or attribute or boolean))) same lock$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/01 15:44
S5	6	(database and index and ((exclusive$2 near3 lock$3) or x?lock) and ((shared near3 lock$3) or s?lock) and ((pseudo?delet$3 or pseudodelet$3 or (pseudo adj delet$3) or (soft adj delet$3) or (delet$3 near2 (flag or flagged or flagging or bit or bool or attribute or boolean))) same lock$3)) and @ad<="20010628"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/01 15:47

S6	50	(search$3 with index) and database and (pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft adj delet$3) or (delet$3 near2 (flag or flagged or flagging or bit or attribute or bool or boolean))) and (lock$3 or x?lock$3 or s?lock$3 or (exclusive$2 adj lock$3) or (shared adj lock$3))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/01 15:50
S7	2	((search$3 with index) and database and (pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft adj delet$3) or (delet$3 near2 (flag or flagged or flagging or bit or attribute or bool or boolean))) and (lock$3 or x?lock$3 or s?lock$3 or (exclusive$2 adj lock$3) or (shared adj lock$3))) and (conditional with (lock$3 or x?lock$3 or s?lock$3))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/01 15:52
S8	45	((search$3 with index) and database and (pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft adj delet$3) or (delet$3 near2 (flag or flagged or flagging or bit or attribute or bool or boolean))) and (lock$3 or x?lock$3 or s?lock$3 or (exclusive$2 adj lock$3) or (shared adj lock$3))) and key	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/01 15:52
S9	40	(((search$3 with index) and database and (pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft adj delet$3) or (delet$3 near2 (flag or flagged or flagging or bit or attribute or bool or boolean))) and (lock$3 or x?lock$3 or s?lock$3 or (exclusive$2 adj lock$3) or (shared adj lock$3))) and key) and @ad<="20010628"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/01 15:53
S10	9	("4914569" "5119490" "5280612" "5551027" "5960194" "6009425" "6032216" "6047285" "6105025"). did.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 12:27
S11	3699	pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 12:29

S12	17	(pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and (x?lock or xlock or (exclusive$2 near1 lock$3)) and (s?lock or slock or (shared near1 lock$3))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 12:32
S13	872	(pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and lock$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 12:32
S14	7	((pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and (x?lock or xlock or (exclusive$2 near1 lock$3)) and (s?lock or slock or (shared near1 lock$3))) and ((unconditional or condition$4) near5 (lock$3 or s?lock or slock))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 12:33
S15	7	(((pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and (x?lock or xlock or (exclusive$2 near1 lock$3)) and (s?lock or slock or (shared near1 lock$3))) and ((unconditional or condition$4) near5 (lock$3 or s?lock or slock))) and @ad<="20010628"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 13:05
S16	51	((pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and lock$3) and (condition$4 with (lock$3 or x?lock or s?lock))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 13:06

S17	16	(((pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and lock$3) and (condition$4 with (lock$3 or x?lock or s?lock))) and (database or data?base)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 13:06
S18	27	(((pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and lock$3) and (condition$4 with (lock$3 or x?lock or s?lock))) and index$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 13:06
S19	34	((((pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and lock$3) and (condition$4 with (lock$3 or x?lock or s?lock))) and (database or data?base)) or ((((pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and lock$3) and (condition$4 with (lock$3 or x?lock or s?lock))) and index$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 13:07

S20	33	(((((pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and lock$3) and (condition$4 with (lock$3 or x?lock or s?lock))) and (database or data?base)) or ((((pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft near1 delet$3) or ((flag or flagging or flagged or mark or marked or marking or bit or bool or boolean) near2 delet$3)) and lock$3) and (condition$4 with (lock$3 or x?lock or s?lock))) and index$3)) and @ad<="20010628"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 13:07
S21	4726	(707/8 707/200 707/201 707/1). ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 13:56
S22	1027	((707/8 707/200 707/201 707/1). ccls.) and (lock$3 or x?lock or s?lock)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 13:56
S23	515	(((707/8 707/200 707/201 707/1). ccls.) and (lock$3 or x?lock or s?lock)) and index$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 13:57
S24	286	((((707/8 707/200 707/201 707/1).ccls.) and (lock$3 or x?lock or s?lock)) and index$3) and transact$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 13:58
S25	37	(((((707/8 707/200 707/201 707/1).ccls.) and (lock$3 or x?lock or s?lock)) and index$3) and transact$3) and (pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft adj delet$3) or ((flag or flagged or flagging or mark or marked or marking or bit or bool or boolean) near2 delet$3))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 14:00

S26	35	((((((707/8 707/200 707/201 707/1).ccls.) and (lock$3 or x?lock or s?lock)) and index$3) and transact$3) and (pseudodelet$3 or pseudo?delet$3 or (pseudo adj delet$3) or (soft adj delet$3) or ((flag or flagged or flagging or mark or marked or marking or bit or bool or boolean) near2 delet$3))) and @ad<="20010628"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2003/08/05 14:00
S27	12	((exclusive$2 near1 lock$3) or x?lock or xlock) and (lock with (flag or attribute or bit!) with delet$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/01/23 11:43
S28	1	shared same (conditional with lock$3 with delet$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/01/23 11:44
S29	1	shared same (conditional near3 lock$3) same delet$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/01/23 11:44
S30	5	((shared with lock$3) or s?lock or slock) and (conditional near3 lock$3) same delet$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/01/23 11:45
S31	5	((shared with lock$3) or s?lock or slock) and ((conditional near3 lock$3) same delet$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/01/23 11:45

Dialog DataStar

options | logoff | feedback | help

databases | easy search

Advanced Search: INSPEC - 1969 to date (INZZ)

limit

Search history:

No.	Database	Search term	Info added since	Results	
1	INZZ	deadlock AND exclusiv$4 AND (pseudodelet$3 OR pseudo-delet$3 OR (soft OR logical OR bit OR attribute OR flag) NEAR delet$3)	unrestricted	0	-
2	INZZ	lock$3 NEAR delet$3 NEAR (attribute OR flag OR bit)	unrestricted	0	-
3	INZZ	lock$3 AND deadlock$3 AND delet$3	unrestricted	3	show titles
4	INZZ	lock$3 AND (pseudodelet$3 OR pseudo-delet$3 OR (soft OR logical OR attribute OR bit OR flag) NEAR delet$3)	unrestricted	1	show titles
5	INZZ	(lock$3 OR concurren$4) AND (deadlock$3 OR dead-lock$3) AND (pseudodelet$3 OR pseudo-delet$3 OR (soft OR logical OR bit OR flag OR attribute) NEAR delet$3)	unrestricted	0	-

hide | delete all search steps... | delete individual search steps...

Enter your search term(s): Search tips

[ ] whole document

Information added since: [ ] or: none
(YYYYMMDD)

search

Select special search terms from the following list(s):

- Classification codes A: Physics, 0-1
- Classification codes A: Physics, 2-3
- Classification codes A: Physics, 4-5
- Classification codes A: Physics, 6
- Classification codes A: Physics, 7
- Classification codes A: Physics, 8

- Classification codes A: Physics, 9
- Classification codes B: Electrical & Electronics, 0-5
- Classification codes B: Electrical & Electronics, 6-9
- Classification codes C: Computer & Control
- Classification codes D: Information Technology
- Classification codes E: Manufacturing & Production
- Treatment codes
- INSPEC sub-file
- Publication types
- Language of publication

Top - News & FAQS - Dialog

© **2005** Dialog

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: [✓ The ACM Digital Library] The Guide

+deadlock +pseudodelet* +exclusive +lock*

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **deadlock** **pseudodelet** **exclusive** **lock** Found **1** of **151,219**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 1 of 1

Relevance scale

1 Research sessions: indexing and tuning: Transaction support for indexed summary views
Goetz Graefe, Michael Zwilling
June 2004 **Proceedings of the 2004 ACM SIGMOD international conference on Management of data**
Full text available: pdf(168.70 KB) Additional Information: full citation, abstract, references

Materialized views have become a standard technique for performance improvement in decision support databases and for a variety of monitoring purposes. In order to avoid inconsistencies and thus unpredictable query results, materialized views and their indexes should be maintained immediately within user transaction just like indexes on ordinary tables. Unfortunately, the smaller a materialized view is, the higher the concurrency contention between queries and updates as well as among concurrent ...

Results 1 - 1 of 1

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+deadlock +delet* +exclusive +lock* +conditional +shared +i

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
deadlock delet exclusive lock conditional shared flag

Found **56** of **151,219**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 56

Result page: **1** 2 3 next

Relevance scale

1 Third Generation Computer Systems
Peter J. Denning
December 1971 **ACM Computing Surveys (CSUR)**, Volume 3 Issue 4

Full text available: pdf(3.62 MB) Additional Information: full citation, abstract, references, citings, index terms

The common features of third generation operating systems are surveyed from a general view, with emphasis on the common abstractions that constitute at least the basis for a "theory" of operating systems. Properties of specific systems are not discussed except where examples are useful. The technical aspects of issues and concepts are stressed, the nontechnical aspects mentioned only briefly. A perfunctory knowledge of third generation systems is presumed.

2 Adaptable concurrency control for atomic data types
M. S. Atkins, M. Y. Coady
August 1992 **ACM Transactions on Computer Systems (TOCS)**, Volume 10 Issue 3

Full text available: pdf(2.54 MB) Additional Information: full citation, abstract, references, index terms, review

In many distributed systems concurrent access is required to a shared object, where abstract object servers may incorporate type-specific properties to define consistency requirements. Each operation and its outcome is treated as an event, and conflicts may occur between different event types. Hence concurrency control and synchronization are required at the granularity of conflicting event types. With such a fine granularity of locking, the occurrence of conflicts is likely to be lower tha ...

Keywords: concurrent access to shared data, hybrid locking, optimistic locking, pessimistic locking, transactions serializability

3 SPLASH: Stanford parallel applications for shared-memory
Jaswinder Pal Singh, Wolf-Dietrich Weber, Anoop Gupta
March 1992 **ACM SIGARCH Computer Architecture News**, Volume 20 Issue 1

Full text available: pdf(3.04 MB) Additional Information: full citation, abstract, citings, index terms

We present the Stanford Parallel Applications for Shared-Memory (SPLASH), a set of parallel applications for use in the design and evaluation of shared-memory multiprocessing systems. Our goal is to provide a suite of realistic applications that will serve as a well-documented and consistent basis for evaluation studies. We describe the applications currently in the suite in detail, discuss some of their important characteristics, and explore their behavior by running them on a real multiprocess ...

4 On randomization in sequential and distributed algorithms
Rajiv Gupta, Scott A. Smolka, Shaji Bhaskar
March 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 1

Full text available: pdf(8.01 MB) Additional Information: full citation, abstract, references, citings, index terms

Probabilistic, or randomized, algorithms are fast becoming as commonplace as conventional deterministic algorithms. This survey presents five techniques that have been widely used in the design of randomized algorithms. These techniques are illustrated using 12 randomized algorithms—both sequential and distributed— that span a wide range of applications, including:primality testing (a classical problem in number theory), interactive probabilistic proof s ...

Keywords: Byzantine agreement, CSP, analysis of algorithms, computational complexity, dining philosophers problem, distributed algorithms, graph isomorphism, hashing, interactive probabilistic proof systems, leader election, message routing, nearest-neighbors problem, perfect hashing, primality testing, probabilistic techniques, randomized or probabilistic algorithms, randomized quicksort, sequential algorithms, transitive tournaments, universal hashing

5 Adaptive and efficient abortable mutual exclusion
Prasad Jayanti
July 2003 **Proceedings of the twenty-second annual symposium on Principles of distributed computing**

Full text available: pdf(1.21 MB) Additional Information: full citation, abstract, references, citings, index terms

Scott and Scherer recently pointed out that existing locking algorithms do not meet a need that arises in practical systems. Specifically, database systems and real time systems need mutual exclusion locks that support the *abort* capability, which makes it possible for a process that waits "too long" to abort its attempt to acquire the lock. Further, to ensure high performance in cache coherent and NUMA multiprocessors, the locking algorithm should generate as few remote references as poss ...

6 Concurrent search structure algorithms
Dennis Shasha, Nathan Goodman
March 1988 **ACM Transactions on Database Systems (TODS)**, Volume 13 Issue 1

Full text available: pdf(2.72 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A dictionary is an abstract data type supporting the actions member, insert, and delete. A search structure is a data structure used to implement a dictionary. Examples include B trees, hash structures, and unordered lists. Concurrent algorithms on search structures can achieve more parallelism than standard concurrency control methods would suggest, by exploiting the fact that many different search structure states represent one dictionary state. We present a framework for verifying such a ...

7 UIO: a uniform I/O system interface for distributed systems
David R. Cheriton
January 1987 **ACM Transactions on Computer Systems (TOCS)**, Volume 5 Issue 1

Full text available: pdf(3.20 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A uniform I/O interface allows programs to be written relatively independently of specific I/O services and yet work with a wide variety of the I/O services available in a distributed environment. Ideally, the interface provides this uniform access without excessive complexity in the interface or loss of performance. However, a uniform interface does not arise from careful design of individual system interfaces alone; it requires explicit definition. In this paper, the UIO (unifo ...

8 Investigations in tree locking for compiled database applications
Heng Yu, Grant E. Weddell
October 2004 **Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research**
Full text available: pdf(273.86 KB) Additional Information: full citation, abstract, references, index terms

We report on initial research in tree locking (TL) schemes for compiled database applications. Such applications have a repository style of architecture in which a collection of software modules operate on a common database in terms of a set of predefined transaction types, an architectural view that is also useful for embedded control programs. Since TL schemes are deadlock free, it becomes possible to entirely decouple concurrency control from any functionality relating to recovery. This pr ...

9 Effective fine-grain synchronization for automatically parallelized programs using optimistic synchronization primitives
Martin C. Rinard
November 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 4
Full text available: pdf(637.69 KB) Additional Information: full citation, abstract, references, citings, index terms, review

This article presents our experience using optimistic synchronization to implement fine-grain atomic operations in the context of a parallelizing compiler for irregular, object-based computations. Our experience shows that the synchronization requirements of these programs differ significantly from those of traditional parallel computations, which use loop nests to access dense matrices using affine access functions. In addition to coarse-grain barrier synchronization, our irregular comput ...

Keywords: atomic operations commutativity analysis, optimistic synchronization, parallel computing, parallelizing compilers, synchronization

10 Using histories to implement atomic objects
Tony P. Ng
November 1989 **ACM Transactions on Computer Systems (TOCS)**, Volume 7 Issue 4
Full text available: pdf(2.74 MB) Additional Information: full citation, abstract, references, citings, index terms, review

In this paper we describe an approach to implementing atomicity. Atomicity requires that computations appear to be all-or-nothing and executed in a serialization order. The approach we describe has three characteristics. First, it utilizes the semantics of an application to improve concurrency. Second, it reduces the complexity of application-dependent synchronization code by analyzing the process of writing it. Third, our approach hides the protocol used to arrive at a ser ...

11 Effective fine-grain synchronization for automatically parallelized programs using optimistic synchronization primitives
Martin Rinard

June 1997 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 32 Issue 7

Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

As shared-memory multiprocessors become the dominant commodity source of computation, parallelizing compilers must support mainstream computations that manipulate irregular, pointer-based data structures such as lists, trees and graphs, Our experience with a parallelizing compiler for this class of applications shows that their synchronization requirements differ significantly from those of traditional parallel computations. Instead of coarse-grain barrier synchronization, irregular computations ...

12 Modern concurrency abstractions for C#

Nick Benton, Luca Cardelli, Cédric Fournet

September 2004 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 26 Issue 5

Full text available: pdf(260.10 KB) Additional Information: full citation, abstract, references, citings, index terms

Polyphonic C&sharp; is an extension of the C&sharp; language with new asynchronous concurrency constructs, based on the join calculus. We describe the design and implementation of the language and give examples of its use in addressing a range of concurrent programming problems.

Keywords: Asynchrony, chords, events, join calculus, messages, polyphonic C&sharp;, synchronization, threads

13 Mostly lock-free malloc

Dave Dice, Alex Garthwaite

June 2002 **ACM SIGPLAN Notices , Proceedings of the 3rd international symposium on Memory management**, Volume 38 Issue 2 supplement

Full text available: pdf(609.93 KB) Additional Information: full citation, abstract, references, citings, index terms

Modern multithreaded applications, such as application servers and database engines, can severely stress the performance of user-level memory allocators like the ubiquitous malloc subsystem. Such allocators can prove to be a major scalability impediment for the applications that use them, particularly for applications with large numbers of threads running on high-order multiprocessor systems.This paper introduces Multi-Processor Restartable Critical Sections, or MP-RCS. MP-RCS permits user-level ...

Keywords: affinity, locality, lock-free operations, malloc, restartable critical sections

14 A proposal for certain process management and intercommunication primitives

Gary D. Knott

October 1974 **ACM SIGOPS Operating Systems Review**, Volume 8 Issue 4

Full text available: pdf(2.52 MB) Additional Information: full citation, references, citings

15 VM-based shared memory on low-latency, remote-memory-access networks

Leonidas Kontothanassis, Galen Hunt, Robert Stets, Nikolaos Hardavellas, Michał Cierniak, Srinivasan Parthasarathy, Wagner Meira, Sandhya Dwarkadas, Michael Scott

May 1997 **ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture**, Volume 25 Issue 2

Full text available: pdf(1.96 MB) Additional Information: full citation, abstract, references, citings, index terms

Recent technological advances have produced network interfaces that provide users with very low-latency access to the memory of remote machines. We examine the impact of such networks on the implementation and performance of software DSM. Specifically, we compare two DSM systems---Cashmere and TreadMarks---on a 32-processor DEC Alpha cluster connected by a Memory Channel network.Both Cashmere and TreadMarks use virtual memory to maintain coherence on pages, and both use lazy, multi-writer releas ...

16 Semaphores revisited with MMS
Pierre Castori
July 1995 **ACM SIGOPS Operating Systems Review**, Volume 29 Issue 3
Full text available: pdf(1.19 MB) Additional Information: full citation, abstract, index terms

The objective of this article is to realize an in-depth study of MMS semaphores. This article is not a tutorial on MMS semaphores although it does explain to some extent how MMS semaphores behave and can be utilized. What we are trying to achieve is a definitive clarification of the MMS semaphore concept in order to make its use easier to MMS programmers and to encourage its adoption in the industrial applications and products. We compare MMS semaphores to Dijkstra's classical semaphores and sho ...

Keywords: MMS, resource, semaphores, service, synchronization

17 Parallel execution of prolog programs: a survey
Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo
July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4
Full text available: pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

18 A partially deadlock-free typed process calculus
Naoki Kobayashi
March 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 20 Issue 2
Full text available: pdf(562.16 KB) Additional Information: full citation, references, citings, index terms

Keywords: concurrency, deadlock-freedom, type theory

19 Semaphore primitives and starvation-free mutual exclusion
Eugene W. Stark
October 1982 **Journal of the ACM (JACM)**, Volume 29 Issue 4

Full text available: pdf(1.40 MB) Additional Information: full citation, references, citings, index terms

20 A Survey of Some Theoretical Aspects of Multiprocessing
J. L. Baer
January 1973 **ACM Computing Surveys (CSUR)**, Volume 5 Issue 1
Full text available: pdf(4.05 MB) Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 56 Result page: **1** 2 3 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player